

# VOICE DATA PROCESSOR AND MEDIUM RECORDING DATA PROCESSING PROGRAM

Publication number: JP10319955

Publication date: 1998-12-04

Inventor: FUKADA AYUMI; MIZUNO SHIGEHIKO

Applicant: YAMAHA CORP

Classification:

- international: G10L13/00; G10H1/00; G10L13/04; G10L13/00;  
G10H1/00; (IPC1-7): G10H1/00; G10H1/00; G10L3/00;  
G10L5/02

- european:

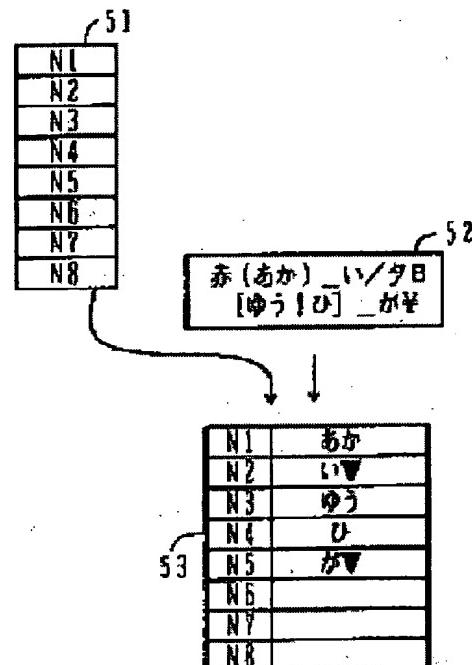
Application number: JP19970148522 19970522

Priority number(s): JP19970148522 19970522

[Report a data error here](#)

## Abstract of JP10319955

**PROBLEM TO BE SOLVED:** To make it possible to process text data including KANA (Japanese syllabary), etc., by retrieving plural kinds of characters as the delimitation data indicating the delimitation of the character strings from the character strings included in text data, allocating the character strings delimited by the retrieved characters to the data of one note in note data to from voice data. **SOLUTION:** Import processing allocates the text data 52 to playing data 51 and forms note plus data 53. The playing data 51 has note events N1 to N8 corresponding to eight notes. The text data 52 is broken down to text events according to text delimitation signals and is allocated to the respective note events N1 to N8. The note plus data 53 deletes KANJI (Chinese characters) by taking read KANA and rubies into consideration and leaves only the KANA corresponding thereto. Namely, the processing of the text data 52 including the KANA or KANJI, etc., is made possible and, therefore, the input or edition of the text data 52 is simple.



4

(19)日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-319955

(43)公開日 平成10年(1998)12月4日

(51) Int.Cl.  
 G 10 H 1/00  
 G 10 L 3/00  
 5/02

識別記号  
 102

F I  
 G 10 H 1/00  
 G 10 L 3/00  
 5/02

Z  
 102 Z  
 J  
 J

審査請求 未請求 請求項の数18 FD (全 22 頁)

(21)出願番号 特願平9-148522

(22)出願日 平成9年(1997)5月22日

(71)出願人 000004075  
 ヤマハ株式会社  
 静岡県浜松市中沢町10番1号

(72)発明者 深田 アユミ  
 静岡県浜松市中沢町10番1号 ヤマハ株式  
 会社内

(72)発明者 水野 成彦  
 静岡県浜松市中沢町10番1号 ヤマハ株式  
 会社内

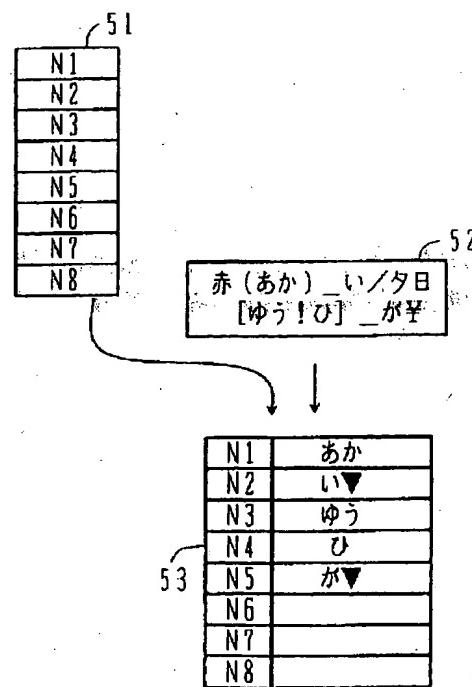
(74)代理人 弁理士 高橋 敬四郎 (外2名)

(54)【発明の名称】 音声データ処理装置及びデータ処理プログラムを記録した媒体

## (57)【要約】

【課題】 音声データの編集を容易に行うことができる  
音声データ処理装置を提供することを課題とする。

【解決手段】 文字列を含むテキストデータ(52)  
と、音符単位のデータを含むノートデータ(51)とを  
記憶する記憶手段と、複数種類の文字を、文字列の区切  
りを示す区切りデータ(\_\_\_\_\_, !)として、前記テキスト  
データ中に含まれる文字列の中から検索し、該検索され  
た文字で区切られる文字列を、前記ノートデータ中の1  
音符のデータに割り当てて音声データ(53)を生成す  
る生成手段とを有する。



**【特許請求の範囲】**

【請求項1】 文字列を含むテキストデータと、音符単位のデータを含むノートデータとを記憶する記憶手段と、複数種類の文字を、文字列の区切りを示す区切りデータとして、前記テキストデータ中に含まれる文字列の中から検索し、該検索された文字で区切られる文字列を、前記ノートデータ中の1音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置。

【請求項2】 文字列を含むテキストデータと、音符単位のデータを含むノートデータとを記憶する記憶手段と、複数種類の文字を、呼気記号として、前記テキストデータ中に含まれる文字列の中から検索し、該検索された文字で区切られる文字列に呼気マークを付加し、前記ノートデータ中の1音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置。

【請求項3】 非仮名とその読み仮名又はルビが対になったデータを含むテキストデータと音符単位のデータを含むノートデータとを記憶する記憶手段と、前記テキストデータ中の非仮名を破棄してその読み仮名又はルビのみを前記ノートデータ中の音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置。

【請求項4】 テキストデータとノートデータとが組になった音声データを記憶する記憶手段と、前記テキストデータとノートデータとの組み合わせを変更して書き込みを行う変更手段とを有する音声データ処理装置。

【請求項5】 複数の文字からなる文字列を含むテキストデータと音符単位のデータを含むノートデータを記憶する記憶手段と、前記テキストデータ中の1文字を前記ノートデータ中の1音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置。

【請求項6】 第1の文字列データと第1のノートデータとが組になった第1のデータとそれに続く第2の文字列データと第2のノートデータとが組になった第2のデータを記憶する記憶手段と、

前記第1の文字列データ中の分割位置を指定する指定手段と、

前記第1の文字列データを前記分割位置で2つに分割し、後部の文字列データを前記第2の文字列データに併合する処理手段とを有する音声データ処理装置。

【請求項7】 文字データとノートデータとが組になった音声データを記憶する記憶手段と、

前記文字データが漢字、英数字又は記号であるときには該文字データを発音するための仮名の候補を提示する提示手段と、

前記仮名の候補の中から1つの仮名を指定し、前記文字データを該仮名に変更して前記記憶手段に記憶させる変更手段とを有する音声データ処理装置。

【請求項8】 歌詞データとノートデータとが組になった形式の音声データを記憶する記憶手段と、ノートデータの微小タイミング前に該ノートデータと組になった歌詞データが位置するように、前記音声データを変換して、歌詞データとノートデータとが独立した形式の音声データとして記録する記録手段とを有する音声データ処理装置。

【請求項9】 歌詞データとノートデータとが独立した形式の音声データであって、ノートデータの微小タイミング前に該ノートデータと組になった歌詞データが位置する音声データを記憶する記憶手段と、

前記音声データを記憶手段から読み出して、前記歌詞データとノートデータとを組にした形式の音声データを生成する生成手段とを有する音声データ処理装置。

【請求項10】 文字列を含むテキストデータと、音符単位のデータを含むノートデータを処理するコンピュータプログラムの記録媒体であって、複数種類の文字を、文字列の区切りを示す区切りデータとして、前記テキストデータ中に含まれる文字列の中から検索し、該検索された文字で区切られる文字列を、前記ノートデータ中の1音符のデータに割り当てて音声データを生成する手順をコンピュータに実行させるためのプログラムを記録した媒体。

【請求項11】 文字列を含むテキストデータと、音符単位のデータを含むノートデータを処理するコンピュータプログラムの記録媒体であって、

複数種類の文字を、呼気記号として、前記テキストデータ中に含まれる文字列の中から検索し、該検索された文字で区切られる文字列に呼気マークを付加し、前記ノートデータ中の1音符のデータに割り当てて音声データを生成する手順をコンピュータに実行させるためのプログラムを記録した媒体。

【請求項12】 非仮名とその読み仮名又はルビが対になったデータを含むテキストデータと音符単位のデータを含むノートデータを処理するコンピュータプログラムの記録媒体であって、

前記テキストデータ中の非仮名を破棄してその読み仮名又はルビのみを前記ノートデータ中の音符のデータに割り当てて音声データを生成する手順をコンピュータに実行させるためのプログラムを記録した媒体。

【請求項13】 (a) テキストデータとノートデータとが組になった音声データを読み出す手順と、

(b) 前記テキストデータとノートデータとの組み合わせを変更して書き込みを行う手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項14】 (a) 複数の文字からなる文字列を含むテキストデータと音符単位のデータを含むノートデータ

タを読み出す手順と、

(b) 前記テキストデータ中の1文字を前記ノートデータ中の1音符のデータに割り当てて音声データを生成する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項15】 第1の文字列データと第1のノートデータとが組になった第1のデータとそれに続く第2の文字列データと第2のノートデータとが組になった第2のデータを処理するコンピュータプログラムの記録媒体であって、

(a) 前記第1の文字列データ中の分割位置を特定する手順と、

(b) 前記第1の文字列データを前記分割位置で2つに分割し、後部の文字列データを前記第2の文字列データに併合する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項16】 文字データとノートデータとが組になった音声データを処理するコンピュータプログラムの記録媒体であって、

(a) 前記文字データが漢字、英数字又は記号であるときには該文字データを発音するための仮名の候補を提示する手順と、

(b) 前記仮名の候補の中から1つの仮名を指定し、前記文字データを該仮名に変更して書き込みを行う手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項17】 (a) 歌詞データとノートデータとが組になった音声データを読み出す手順と、

(b) ノートデータの微小タイミング前に該ノートデータと組になった歌詞データが位置するように、前記音声データを変換して、歌詞データとノートデータとが独立した形式の音声データとして記録する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

【請求項18】 (a) 歌詞データとノートデータとが独立した形式の音声データであって、該ノートデータの微小タイミング前に該ノートデータと組になった歌詞データが位置する音声データを読み出す手順と、

(b) 前記音声データを基に前記歌詞データとノートデータとを組にした形式の音声データを生成する手順とをコンピュータに実行させるためのプログラムを記録した媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】 本発明は、音声データ処理技術に関し、特に音声発音装置に音声を発音させるための音声データの処理技術に関する。

##### 【0002】

【従来の技術】 音声発音装置に供給する音声データの処理を行うことができる音声データ処理装置が知られている。音声発音装置は、例えばフォルマント音源等の音声

合成回路を備えている。フォルマント音源は、音声を周波数解析することにより形成されるフォルマントを合成することにより音声信号を生成する。

【0003】 音節は、いわゆる50音のうちの1音等であり、例えば「か」である。音節は、時間軸上の音素列に分解することができる。例えば、音節「か」は、音素列「CL(7.5ms) + kha(4×7.5ms) + aj(無限長)」に分解することができる。音素は、例えば「CL」、「kha」、「aj」である。フォルマント音源は、音素列を入力することにより音声信号を生成する。

【0004】 従来技術による音声データ処理装置として、特開平9-50287号公報や特開平9-44179号公報が知られている。これらは、メロディデータと歌詞データをメモリに記憶し、両方のデータをメモリから順次読み出し、フォルマント音源へ供給する。フォルマント音源は、データを受けて歌声を発生する。

【0005】 データ処理装置として、音節の編集を行うことができるワードプロセッサが知られている。しかし、音素の編集を行うことができるデータ処理装置がないため、音声データはマニュアル設定を行っていた。

##### 【0006】

【発明が解決しようとする課題】 特開平9-50287号公報のデータ処理装置は、歌詞データをローマ字表記で入力する必要があり、仮名又は漢字等を使うことができないため、歌詞データの入力又は編集が不便である。例えば、仮名又は漢字が混在した既存の歌詞データを使用することができない。

【0007】 また、各音符に歌詞データを割り当てるため、歌詞データ中に音符の区切り（歌詞区切り）記号を設ける必要があるが、歌詞区切りは1つの記号（例えばスペース「\_」）のみに限定されている。そのため、改行マーク（歌詞表示画面上において改行をする位置を示すマーク）と副改行マーク（通常は利用されないが、表示スペースが小さい等の表示器の都合により前記改行マークまでの歌詞が1行に入りきらない場合に、改行マークよりも前で改行する位置を示すマーク）の2種類の改行マークが含まれているカラオケ用の歌詞データにおいて、改行マークのみを歌詞区切りとして用いることは可能であっても、改行マークと副改行マークの両方を歌詞区切りとして用いることはできない。

【0008】 さらに、歌詞データ等は、独自のデータフォーマットを有しており、歌詞データの再生装置として専用の装置が必要となり、一般的なシーケンサ等での再生ができない。

【0009】 本発明の目的は、仮名等を含むテキストデータを処理することができる音声データ処理装置又はコンピュータプログラムの記録媒体を提供することである。

【0010】 本発明の他の目的は、音声データの編集を

容易に行うことができる音声データ処理装置又はコンピュータプログラムの記録媒体を提供することである。

【0011】本発明の他の目的は、汎用性のある音声データを処理することができる音声データ処理装置又はコンピュータプログラムの記録媒体を提供することである。

#### 【0012】

【課題を解決するための手段】本発明の一観点によれば、文字列を含むテキストデータと、音符単位のデータを含むノートデータとを記憶する記憶手段と、複数種類の文字を、文字列の区切りを示す区切りデータとして、前記テキストデータ中に含まれる文字列の中から検索し、該検索された文字で区切られる文字列を、前記ノートデータ中の1音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置が提供される。

【0013】本発明の他の観点によれば、文字列を含むテキストデータと、音符単位のデータを含むノートデータとを記憶する記憶手段と、複数種類の文字を、呼気記号として、前記テキストデータ中に含まれる文字列の中から検索し、該検索された文字で区切られる文字列に呼気マークを付加し、前記ノートデータ中の1音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置。が提供される。

【0014】本発明の他の観点によれば、非仮名とその読み仮名又はルビが対になったデータを含むテキストデータと音符単位のデータを含むノートデータとを記憶する記憶手段と、前記テキストデータ中の非仮名を破棄してその読み仮名又はルビのみを前記ノートデータ中の音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置が提供される。

【0015】本発明の他の観点によれば、テキストデータとノートデータとが組になった音声データを記憶する記憶手段と、前記テキストデータとノートデータとの組み合わせを変更して書き込みを行う変更手段とを有する音声データ処理装置が提供される。

【0016】本発明の他の観点によれば、複数の文字からなる文字列を含むテキストデータと音符単位のデータを含むノートデータを記憶する記憶手段と、前記テキストデータ中の1文字を前記ノートデータ中の1音符のデータに割り当てて音声データを生成する生成手段とを有する音声データ処理装置が提供される。

【0017】本発明の他の観点によれば、第1の文字列データと第1のノートデータとが組になった第1のデータとそれに続く第2の文字列データと第2のノートデータとが組になった第2のデータを記憶する記憶手段と、前記第1の文字列データ中の分割位置を指定する指定手段と、前記第1の文字列データを前記分割位置で2つに分割し、後部の文字列データを前記第2の文字列データに併合する処理手段とを有する音声データ処理装置が提

供される。

【0018】本発明の他の観点によれば、文字データとノートデータとが組になった音声データを記憶する記憶手段と、前記文字データが漢字、英数字又は記号であるときには該文字データを発音するための仮名の候補を提示する提示手段と、前記仮名の候補の中から1つの仮名を指定し、前記文字データを該仮名に変更して前記記憶手段に記憶させる変更手段とを有する音声データ処理装置が提供される。

【0019】本発明の他の観点によれば、歌詞データとノートデータとが組になった形式の音声データを記憶する記憶手段と、ノートデータの微小タイミング前に該ノートデータと組になった歌詞データが位置するように、前記音声データを変換して、歌詞データとノートデータとが独立した形式の音声データとして記録する記録手段とを有する音声データ処理装置が提供される。

【0020】本発明の他の観点によれば、歌詞データとノートデータとが独立した形式の音声データであって、ノートデータの微小タイミング前に該ノートデータと組になった歌詞データが位置する音声データを記憶する記憶手段と、前記音声データを記憶手段から読み出して、前記歌詞データとノートデータとを組にした形式の音声データを生成する生成手段とを有する音声データ処理装置が提供される。

#### 【0021】

【発明の実施の形態】図1は、パーソナルコンピュータ1と外部音源装置21との接続を示す図である。パーソナルコンピュータ1は、本実施例による音声データ処理装置を含む。パーソナルコンピュータ1は、シーケンサでもよい。

【0022】まず、外部音源装置21の構成を説明する。検出回路33は、スイッチ34の操作を検出し、スイッチ信号を生成する。スイッチ34は、例えば各種パラメータを設定するためのスイッチを含む。

【0023】バス22には、検出回路33の他、RAM24、ROM25、CPU26、表示回路28、MIDIインターフェース23、音声合成回路32、楽音波形合成回路29が接続される。

【0024】ROM25は、音声を合成するためのフォルマントデータ、その他の各種データ及びコンピュータプログラムを記憶する。RAM24は、フラグ、バッファ等を記憶する。コンピュータプログラムは、ROM25に記憶させる代わりに、RAM24に記憶させてもよい。CPU26は、ROM25又はRAM24に記憶されているコンピュータプログラムに従い、演算または制御を行う。

【0025】CPU26は、MIDIインターフェース23を介してパーソナルコンピュータ1から受信した演奏データに基づいて楽音パラメータを生成し、楽音波形合成回路29に供給する。楽音波形合成回路29は、供

給される楽音パラメータに応じて楽音信号を生成し、サウンドシステム31に供給する。サウンドシステム31は、D/A変換器およびスピーカを含み、供給されるデジタル形式の楽音信号をアナログ形式に変換し、発音する。

【0026】なお、楽音波形合成回路29は、波形メモリ方式、FM方式、物理モデル方式、高調波合成方式、フォルマント合成方式、VCO+VCF+VCAのアナログシンセサイザ方式等、どのような方式であってもよい。

【0027】MIDIインターフェース23は、パソコン用コンピュータ1のMIDIインターフェース8にMIDIケーブルで接続される。外部音源装置21とパソコン用コンピュータ1は、MIDI通信を行うことができる。

【0028】CPU26は、MIDIインターフェース23を介して、パソコン用コンピュータ1から音声データを受け取り、RAM24に格納する。音声データは、ノート(音符)データと歌詞データを含む。

【0029】CPU26は、RAM24に記憶されている音声データを読み出し、ROM25に記憶されているフォルマントデータ等に基づき、フォルマントデータやノートナンバ(音高)等を音声合成回路32に供給する。フォルマントデータは、例えばフォルマント中心周波数データ、フォルマントバンド幅データ、フォルマントレベルデータ等である。

【0030】音声合成回路32は、供給されるフォルマントデータ等に応じて音声信号を生成する。音声信号は、所定の音高を有し、歌声に相当する。音声合成回路32は、フォルマント合成方式(フォルマント音源)でもよいし、その他の方式でもよい。なお、音声合成回路32の方式がフォルマント合成方式以外の場合は、供給するデータもフォルマントデータではなく、その音声合成方式に特有のデータとなる。

【0031】音声合成回路32がフォルマント音源である場合を説明する。音声合成回路32には、音声データに応じて生成されたフォルマントデータ等が入力される。フォルマントデータは、フォルマント中心周波数データ、フォルマントレベルデータ、フォルマント帯域幅データ等を含む。

【0032】音声合成回路32は、有声音合成グループVTGと無声音合成グループUTGを有する。有声音グループVTGは、フォルマントデータ等に応じて、第1～4の有声音フォルマントをそれぞれ形成するための有声音合成部V1、V2、V3、V4を有する。無声音合成グループUTGは、フォルマントデータ等に応じて、第1～4の無声音フォルマントをそれぞれ形成するための無声音合成部U1、U2、U3、U4を有する。

【0033】第1フォルマント生成ユニットTG1は第1フォルマントを形成するための有声音合成部V1と無

声音合成部U1を有し、第2フォルマント生成ユニットTG2は第2フォルマントを形成するための有声音合成部V2と無声音合成部U2を有し、第3フォルマント生成ユニットTG3は第3フォルマントを形成するための有声音合成部V3と無声音合成部U3を有し、第4フォルマント生成ユニットTG4は第4フォルマントを形成するための有声音合成部V4と無声音合成部U4を有する。

【0034】合成部V1～V4、U1～U4は、それぞれ生成したフォルマントに対応する音声信号を出力する。それらの音声信号は加算され、サウンドシステム31に供給される。サウンドシステム31は、供給されるデジタル形式の音声信号をアナログ形式に変換し、発音する。

【0035】フォルマント音源のより具体的な構成は、例えば特開平3-200299号公報の第1図に示されている。

【0036】なお、音声合成回路32は、4系統のフォルマント生成ユニットTG1～TG4を有する場合に限定されない。それ以上又はそれ以下の系統のフォルマント生成ユニットを有するものでもよい。

【0037】破裂音等の特殊な音は、専用のユニットを設けるようにしてもよい。また、発音チャンネルを複数用意し、2声以上の歌唱を発生させることができるようにもよい。この場合、1つの回路を時分割で使用することによって複数の発音チャンネルを形成するようなものでもよいし、1つの発音チャンネルが1つの回路で構成されるような形式のものであってもよい。

【0038】音声合成回路32と楽音波形合成回路29は、専用のハードウェアを用いて構成するものに限らず、DSP+マイクロプログラムを用いて構成してもよいし、CPU+ソフトウェアのプログラムで構成するようにしてもよい。

【0039】次に、パソコン用コンピュータ1の構成を説明する。検出回路11は、キーボード12上のキー(数字キーや文字キー等)入力を検出し、キー信号を生成する。検出回路9は、マウス10の移動操作やスイッチ(右ボタン、左ボタン等)操作を検出し、マウス信号を生成する。操作者は、マウス10又はキーボード12を用いて、音声データの編集を行うことができる。

【0040】表示回路7には、音声データ等を編集するための編集画面を表示することができる。操作者は、表示回路7上の編集画面を参照しながら音声データの編集を行うことができる。

【0041】バス2には、検出回路11、検出回路9及び表示回路7の他、MIDIインターフェース8、RAM3、ROM4、CPU5、外部記憶装置13、通信インターフェース14が接続される。

【0042】ROM4は、各種パラメータ及びコンピュータプログラムを記憶する。RAM3は、フラグ、バッ

ファ、演奏データ、音声データ等を記憶する。また、RAM3は、外部記憶装置13、又は通信インターフェース14を介して外部から供給されたコンピュータプログラムや演奏データ、音声データ等を記憶することもできる。CPU5は、RAM3又はROM4に記憶されているコンピュータプログラムに従い、音声データの編集又は処理等のための演算または制御を行う。

【0043】タイマ6は、CPU5に時間情報を供給する。CPU5は、当該時間情報に応じて、所定時間間隔で割り込み処理を行うことができる。

【0044】MIDIインターフェース8は、上記のように、外部音源装置21のMIDIインターフェース23にMIDIケーブルで接続される。パーソナルコンピュータ1は、MIDIインターフェース8を介して、演奏データや音声データを外部音源装置21に送信することができる。

【0045】通信インターフェース14は、ローカルエリアネットワーク(LAN)、インターネット、電話回線等の通信ネットワーク41に接続される。通信ネットワーク41には、サーバコンピュータ42が接続されている。パーソナルコンピュータ1は、通信ネットワーク41を介して、サーバコンピュータ42から音声データ又はコンピュータプログラムを受信することができる。

【0046】外部記憶装置13は、外部記憶装置用のインターフェースを含み、そのインターフェースを介してバス2に接続される。外部記憶装置13は、例えばフロッピディスクドライブ(FDD)、ハードディスクドライブ(HDD)、光磁気ディスク(MO)ドライブ、CD-ROM(コンパクトディスクリードオンリィメモリ)ドライブ、デジタル多目的ディスク(DVD)装置等である。演奏データや音声データは、外部記憶装置13又はRAM3に記憶される。

【0047】コンピュータプログラム等をROM4に記憶せずに、外部記憶装置13(例えばハードディスク)に記憶させることもできる。ハードディスクからRAM3にコンピュータプログラム等を読み出すことにより、ROM4にコンピュータプログラム等を記憶させている場合と同様の動作をCPU5にさせることができる。このようにすると、CD-ROM等の他の外部記憶媒体からハードディスクにコンピュータプログラム等をコピーすることでコンピュータプログラム等の追加やバージョンアップ等が容易に行える。

【0048】通信インターフェース14は、ローカルエリアネットワーク(LAN)やインターネット、電話回線等の通信ネットワーク41に接続されており、該通信ネットワーク41を介して、サーバコンピュータ42と接続される。外部記憶装置13内にコンピュータプログラム等が記憶されていない場合、サーバコンピュータ42からコンピュータプログラム等をダウンロードすることができる。クライアントとなるパーソナルコンピュー

タ1は、通信インターフェース14及び通信ネットワーク41を介してサーバコンピュータ42へとコンピュータプログラム等のダウンロードを要求するコマンドを送信する。サーバコンピュータ42は、このコマンドを受け、要求されたコンピュータプログラム等を、通信ネットワーク41を介してパーソナルコンピュータ1へと配信し、パーソナルコンピュータ1が通信インターフェース14を介して、これらコンピュータプログラム等を受信して外部記憶装置13内に蓄積することにより、ダウンロードが完了する。

【0049】パーソナルコンピュータ1は、RAM3に記憶している演奏データや音声データを、所定のテンポにしたがって再生し、再生した演奏データや音声データをMIDIインターフェース8を介して外部音源装置21へと送信する。外部音源装置21は、送信されてきた演奏データや音声データをMIDIインターフェース23を介して受信し、受信した演奏データを楽音波形合成回路29へ供給することにより伴奏楽音信号を形成するとともに、受信した音声データを音声合成回路32へ供給することにより歌唱信号を形成する。すなわち、パーソナルコンピュータ1と外部音源装置21により、伴奏楽音を伴った歌唱を生成するシステムが構成される。

【0050】次に、音声データ処理装置(パーソナルコンピュータ1)が行う処理を説明する。

【0051】図2は、インポート処理を示す図である。インポート処理は、演奏データ51にテキストデータ(歌詞データ)52を割り当て、ノートプラスデータ53を生成する処理である。本明細書では、演奏データ51、テキストデータ52、ノートプラスデータ53のいずれをも音声データという。

【0052】演奏データ51は、例えば、8つのノート(音符)に対応するノートイベントN1～N8を有する。ノートイベントN1～N8の各々は、例えばノートオンタイミングデータ、ノートナンバ(音高)、ベロシティ(音量)、ゲートタイム(ノートオンからノートオフまでの時間)を含む。

【0053】テキストデータ52は、例えば「赤(あか)」\_い／夕日〔ゆう！ひ〕\_が￥」である。ここで、〔〕内は読み仮名を示し、〔〕内はルビを示し、「\_！」等は歌詞区切り記号を示し、「／」等は呼氣記号を示す。

【0054】呼氣記号は、息継ぎを示す記号である。呼氣記号があるときは、その前の歌詞データはゲートタイムだけ発音され、その後、無音状態になる。呼氣記号がないときには、ある歌詞データとその次の歌詞データとが滑らかにつながるように発音される。すなわち、ゲートタイムが無視される。

【0055】歌詞区切り記号は、1つの音符(ノートイベント)に対応する歌詞データ(第1の歌詞イベント)とその次の音符に対応する歌詞データ(第2の歌詞イベ

ント) の間に挿入される。歌詞データ 5 2 は、歌詞区切り記号に従い、歌詞イベントに分解され、各ノートイベント N 1～N 8 に割り当てられる。例えば、「N 1. あか」、「N 2. い△」、「N 3. ゆう」、「N 4. ひ」、「N 5. が△」のように割り当てられる。ここで、「△」は、呼気記号「／, ¥」に対応する呼気データであり、息継ぎを示すとともに、「N 2. い△」の例でも判るとおり、歌詞区切りの役割も果たしている。このノートイベントと歌詞(文字列)イベントあるいは音素列イベント(音素列イベントについては後述)の組をノートプラスイベント(ノートに対して歌詞イベントがプラスされたイベント)と呼ぶ。ノートプラスデータ 5 3 は、ノートプラスイベントの集まりである。

【0056】ノートプラスデータ 5 3 は、読み仮名及びルビを考慮し、漢字を削除してそれに対応する仮名のみを残す。読み仮名記号「()」があるときには、()の前の漢字「赤」及び読み仮名記号「()」を削除して()内の仮名「かな」を残す。ルビ記号「[]」の場合も同様である。

【0057】ノートプラスデータ 5 3 中の仮名は、音素列に変換される。例えば、仮名「か」は、音素列「C L (7. 5 ms) + k h a (4 × 7. 5 ms) + a j (無限長)」に変換される。「か」等の音節は、時間軸上の音素列に分解することができる。音素は、例えば「C L」、「k h a」、「a j」である。

【0058】パーソナルコンピュータ 1 は、仮名と音素列との間で変換を行うためのテーブルを記憶している。このテーブルを使い、仮名から音素列への変換を行うことができる。しかし、漢字や記号や英数字等から音素列への変換を行うことはできない。なぜなら、漢字や記号や英数字等は、読み方が複数通り存在する可能性があるからである。

【0059】仮名は音素列へ変換され、漢字や記号等はそのまま残される。パーソナルコンピュータ 1 は、漢字等から仮名へ変換を行うための辞書を記憶している。この辞書を用いて、漢字等を仮名に変換し、さらにその仮名を音素列に変換することができる。パーソナルコンピュータ 1 は、音素列等を外部音源装置 2 1(図 1) に供給する。外部音源装置 2 1 は、音素列等を受けて、音声を発音する。

【0060】パーソナルコンピュータ 1 は、歌詞データとしてテキストデータを使用することができる。例えばカラオケ用のテキストデータ等、既存のテキストデータを使用することができる。

【0061】また、歌詞区切り記号は「\_」、「!」等であり、呼気記号は「/」、「¥」等であり、いずれも複数種類設定することができる。例えば、カラオケ用のテキストデータを使用する場合、改行マークと副改行マークの両方を歌詞区切り記号とし、改ページマークを呼気記号として設定することができる。歌詞区切り記号を複数種類

設定することができるので、改行マークと副改行マークの両方を歌詞区切り記号として使用することができる。

【0062】改行マークは、歌詞を画面に表示する際に、歌詞の 1 行分の区切りを示すマークである。副改行マークは、小さい画面に表示する際にのみ用いる改行のためのマークである。すなわち、通常画面では改行マークのみで改行を行い、小画面では改行マークと副改行マークの両方で改行を行う。例えば、改行マークとして「\_」を使用し、副改行マークとして「!」を使用することができる。なお、歌詞区切り記号や呼気記号は、改行マークや副改行マークのように本来は他の用途に用いるために挿入された記号をこれら歌詞区切り記号や呼気記号としてもよい。

【0063】図 3 は、パーソナルコンピュータ 1 の CPU が行うテキストインポート処理を示すフローチャートである。テキストインポート処理は、上記のように(図 2)、演奏データ 5 1 にテキストデータ 5 2 を割り当てる、ノートプラスデータ 5 3 を生成する処理である。

【0064】ステップ SA 1 では、インポートするテキストファイルを選択する。テキストファイルはテキストデータ 5 2(図 2) を含み、外部記憶装置 1 3(図 1) 等に記憶される。操作者は、マウス 1 0 等を用いて、1 つ又は複数のテキストファイルの中から 1 つを選択することができる。

【0065】ステップ SA 2 では、演奏データ 5 1 中の最初のノートイベント N 1 の位置にポインタをセットする。演奏データ 5 1 は、RAM 4 又は外部記憶装置 1 3(図 1) 等に記憶される。

【0066】ステップ SA 3 では、テキストデータ 5 2 中の最初の文字コードの位置にポインタをセットする。例えば、テキストデータ 5 2 中の「赤」の位置にポインタをセットする。

【0067】ステップ SA 4 では、当該文字コードが呼気記号として設定されている文字コードのいずれかと一致するか否かをチェックする。例えば、呼気記号として、「/」、「¥」が設定されている。一致しないときは、NO の矢印に従い、ステップ SA 5 へ進む。

【0068】ステップ SA 5 では、当該文字コードが歌詞区切り記号として設定されている文字コードのいずれかと一致するか否かをチェックする。例えば、歌詞区切り記号として、「\_」、「!」が設定されている。一致しないときは、NO の矢印に従い、ステップ SA 6 へ進む。

【0069】ステップ SA 6 では、当該文字コードを文字列レジスタの内容に追加する。文字列レジスタは、1 音符に対応する 1 又は複数の文字コード(以下、歌詞イベントという)を記憶するためのレジスタであり、初期時には何も記憶していない。

【0070】ステップSA7では、次の文字コードの位置にポインタをセットする。例えば、テキストデータ52中の「(」の位置にポインタをセットする。その後、ステップSA4へ戻り、上記の処理を繰り返す。

【0071】ステップSA7でセットされたポインタの位置における文字コードが「\_」又は「!」であった場合は、ステップSA4を介して、ステップSA5へ進み、文字コード「\_」又は「!」が歌詞区切り記号であると判断し、YESの矢印に従いステップSA9へ進む。

【0072】一方、ステップSA7でセットされたポインタの位置における文字コードが「/」又は「¥」あつた場合は、ステップSA4で文字コード「/」又は「¥」が呼気記号であると判断し、YESの矢印に従いステップSA8へ進む。ステップSA8では、呼気マーク「▽」を文字列レジスタの内容に追加する。その後、ステップSA9へ進む。

【0073】ステップSA9では、読み仮名及びルビ処理を行う。具体的には、文字列レジスタに格納された歌詞イベント内に読み仮名又はルビが含まれている場合には、漢字等を削除して仮名を残す処理である。この処理の詳細は、後に図4のフローチャートを参照しながら説明する。その後、ステップSA10へ進む。

【0074】ステップSA10では、文字列レジスタの内容をポインタが位置するノートイベントに付加して、ノートプラスイベントを生成する。例えば、文字列レジスタ内の歌詞イベント「あか」をノートイベントN1に付加して、ノートプラスイベント「N1, あか」を生成する。

【0075】ステップSA11では、テキストデータ52の続きがあるか否かをチェックする。すなわち、テキストデータ52が終わりか否かをチェックする。終わりでないときには、YESの矢印に従い、ステップSA12へ進む。

【0076】ステップSA12では、文字列レジスタの内容をクリアし、次の歌詞イベントの格納に備える。

【0077】ステップSA13では、演奏データ51中の次のノートイベントの位置にポインタをセットする。例えば、ノートイベントN2の位置にポインタをセットする。

【0078】ステップSA14では、テキストデータ52中の次の文字コードの位置にポインタをセットする。例えば、文字コード「い」の位置にポインタをセットする。その後、ステップSA4へ戻り、上記の処理を繰り返す。

【0079】テキストデータ52の最後の文字コードの処理が終了すると、ステップSA11でNOの矢印に従い、テキストインポート処理を終了する。この処理により、ノートプラスデータ53が生成される。

【0080】なお、上記のフローチャートでは、テキス

トファイル中のテキストデータ52をインポートする場合を説明したが、スタンダードMIDIファイル中の所定トラックに含まれるテキストをインポートするようにしてもよい。

【0081】図4は、図3のステップSA9に示す読み仮名及びルビ処理の詳細を示すフローチャートである。

【0082】ステップSB1では、文字列レジスタ中に()又は[]に囲まれた仮名の文字コードがあるか否かをチェックする。例えば、テキストデータ52中の「(あか)」又は「[ゆうひ]」がこれに該当する。

【0083】また、「夕日[ゆうひ]」の場合は、文字列「夕日[ゆう]」と文字列「ひ」との両方もこれに該当する。すなわち、()又は[]のいずれかの片括弧に囲まれた仮名文字があればこれに該当する。

【0084】当該文字コードがあるときには、YESの矢印に従い、ステップSB2へ進む。当該文字コードがないときには、NOの矢印に従い、読み仮名及びルビ処理を終了する。

【0085】ステップSB2では、文字列レジスタ中の()又は[]の前にある非仮名の文字コードを削除する。非仮名の文字コードは、漢字、記号、英数字を含む。例えば、テキストデータ52中の「赤」や「夕日」を削除する。

【0086】ステップSB3では、文字列レジスタ中の()又は[]を削除する。その結果、例えば、文字列「夕日[ゆう]」は「ゆう」に変換され、文字列「ひ」」は「ひ」に変換される。その後、読み仮名及びルビ処理を終了し、図3のテキストインポート処理に戻る。

【0087】上記の処理により、読み仮名やルビが振られた漢字等は、漢字等が削除されて仮名が残る。漢字等は、発音に不必要的文字コードである。

【0088】図5は、パーソナルコンピュータ1のCPUが行う音素列変換処理を示すフローチャートである。音素列変換処理は、ノートプラスデータ53(図2)を生成した後に、歌詞イベントを音素列に変換する処理である。

【0089】表示回路7においては、1つのノートプラスイベントが1行に表示されている。すなわち、1つの行は1つのノートプラスイベントに対応する。ステップSC1では、ノートプラスデータ53中から変換する行を選択する。操作者は、表示回路7に表示された各行の表示内容を参照し、マウス10等を用いて、1行又は複数行(すなわち1又は複数のノートプラスイベント)を選択することができる。

【0090】ステップSC2では、選択された行の中の第1行目の位置にポインタをセットする。例えば、ノートプラスイベント「N1, あか」の位置にポインタをセットする。

【0091】ステップSC3では、当該行の文字列(歌詞イベント)は仮名のみか否かをチェックする。仮名の

みのときは、音素列変換を行うため、YESの矢印に従いステップSC4へ進む。仮名のみでないときには、NOの矢印に従い、音素列変換を行わずにステップSC6へ進む。なお、仮名以外に呼気記号も含まれている場合は、仮名のみと判断する。

【0092】ステップSC4では、仮名一音素列テーブルを参照して、当該文字列に対応する音素列を求める。例えば、文字列「あか」に対応する音素列を求める。

【0093】ステップSC5では、文字列と音素列との置き換えを行う。すなわち、ノートプラスデータ53中では、文字列「あか」が削除され、それに対応する音素列が書き込まれる。その後、ステップSC6へ進む。

【0094】ステップSC6では、選択範囲行の中に次の行があるか否かをチェックする。選択範囲行が1行のみの場合は次の行がないので、NOの矢印に従い、音素列変換処理を終了する。次の行があるときには、YESの矢印に従い、ステップSC7へ進む。

【0095】ステップSC7では、次の行の位置にピントをセットする。例えば、ノートプラスイベント「N2, い△」の位置にピントをセットする。その後、ステップSC3へ戻り、上記の処理を繰り返す。呼気記号「△」は音素列に変換されない。仮名「い」のみが音素列に変換される。

【0096】全ての行の処理が終了すると、ステップSC6においてNOの矢印に従い、音素列変換処理を終了する。後の処理で、変換された音素列は、外部音源装置21(図1)へ供給され、発音される。

【0097】なお、音素列に変換された後は、当該音素列を再び文字列に変換して、音素列と文字列の両方を表示回路7上の画面に表示する。操作者は、文字列とそれに対応する音素列を知ることができる。また、音素列の変換が正常に行われたか否かを知ることもできる。

【0098】次に、空白行挿入処理を説明する。上記のように、ノートイベントに歌詞イベントを割り当てた後、歌詞の間に新たな歌詞を挿入したい場合がある。例えば、歌詞の一部分が欠落しており、該欠落部分以降の歌詞が前方に詰まった状態でノートイベントと歌詞イベントが対応づけられてしまい、結果として、ノートイベントと歌詞イベントの対応が間違った状態でソートプラスイベントが生成されてしまった場合等である。その場合に、所望の場所に空白行を挿入し、その空白行に新たな歌詞を埋めればよい。

【0099】図6(A)～(D)は、空白行挿入処理を説明するための図である。図6(A)は、ノートプラスデータの例を示す。ノートプラスデータは、例えば8行のノートプラスイベントを有する。第1行目のノートプラスイベントは、ノートイベントN1と歌詞イベントL1を有する。

【0100】操作者は、マウス等を用いて、空白行を挿入したい位置PTと行数LLを指定することができる。

例えば、挿入位置PTとして第3行目を指定し、挿入行数LLとして2行を指定することができる。

【0101】まず、図6(B)のバッファに、挿入位置PT以降の歌詞イベントL3～L8をコピーする。なお、バッファ内には、空白行が含まれていてもよい。

【0102】次に、図6(C)に示すように、図6(A)のノートプラスデータの挿入位置PT(第3行目)から挿入行数LL(2行)だけ歌詞イベントを消去する。すなわち、歌詞イベントL3及びL4を消去する。

【0103】次に、図6(D)に示すように、図6(C)のノートプラスデータ中の第5行目以降にバッファ(図6(B))内の歌詞イベントをコピーする。ただし、ノートイベントN8よりも後ろにはノートイベントがないので、ノートイベントN8に残りの歌詞イベントがまとめて割り当てられる。すなわち、ノートイベントN8には、歌詞イベントL6～L8が割り当てられる。

【0104】図6(D)のノートプラスデータでは、ノートイベントN3及びN4に空白行の歌詞イベントが割り当てられる。結果として、2行の空白行が挿入されたことになる。そして、空白行挿入箇所以降のノートプラスイベントは、ノートイベントと歌詞イベントの対応づけが変更された状態となる。なお、空白行については、歌詞イベントが消去されたため、ノートプラスイベントではなく、通常のノートイベントとなる。

【0105】図7は、上記の空白行挿入処理を実現するためのフローチャートである。ステップSD1では、空白行の挿入位置PTと挿入行数LLを指定する。

【0106】ステップSD2では、挿入位置PT以降の歌詞イベント(音素列/文字列)をバッファ(図6(B))にコピーする。

【0107】ステップSD3では、挿入位置PTから挿入行数LL分の歌詞イベントを消去する(図6(C))。

【0108】ステップSD4では、バッファ(図6(B))の歌詞イベントを、挿入位置PTから挿入行数LL分より後のノートイベントに割り当てる。このとき、割り当てるべきノートイベントが無くなつたにもかかわらず歌詞イベントがバッファ内に残っているときは、残りの歌詞イベントを全て最後のノートイベントに追加する。以上で、空白行の挿入処理を終了する。

【0109】なお、図6(A)のノートプラスデータに空白行を挿入して図6(D)のノートプラスデータを生成する方法は上記以外の方法でもよい。

【0110】図8(A)～(F)は、歌詞イベント挿入処理を説明するための図である。図8(A)は、ノートプラスデータの例を示す。操作者は、マウス等を用いて、挿入したい歌詞イベントの範囲RGを指定することができる。例えば、範囲RGとして、歌詞イベントL5及びL6を指定することができる。

【0111】まず、図8(B)のクリップボードに、範囲RG内の歌詞イベントL5及びL6をコピーする。

【0112】次に、図8(C)に示すように、操作者は、マウス等を用いて、挿入位置PTを指定することができます。例えば、挿入位置PTとして、第3行目のノートプラスイベント「N3, L3」を指定することができる。

【0113】次に、図8(D)のバッファに、挿入位置PT以降の歌詞イベントL3～L8をコピーする。

【0114】次に、図8(E)に示すように、図8(C)の挿入位置PT以降のノートイベントN3及びN4に、クリップボード(図8(B))の歌詞イベントL5及びL6を割り当てる。

【0115】次に、図8(F)に示すように、図8(E)のノートプラスデータ中の第5行目「N5, L5」以降に、バッファ(図8(D))内の歌詞イベントL3～L8を割り当てる。ただし、ノートイベントN8よりも後ろにはノートイベントがないので、ノートイベントN8に残りの歌詞イベントL6～8をまとめて割り当てる。その結果、図8(F)のノートプラスデータには、挿入位置PTに歌詞イベントL5及びL6が挿入されたことになる。

【0116】図9は、上記の歌詞イベント挿入処理を実現するためのフローチャートである。

【0117】ステップS E 1では、任意行数RGの歌詞イベントを指定し(図8(A))、クリップボード(図8(B))にコピーする。

【0118】ステップS E 2では、歌詞イベントの挿入位置PTを指定する(図8(C))。

【0119】ステップS E 3では、挿入位置PT以降の歌詞イベントをバッファ(図8(D))にコピーする。

【0120】ステップS E 4では、クリップボード(図8(B))の歌詞イベントを、挿入位置PTから挿入行数RG分だけノートプラスデータに割り当てる(図8(E))。

【0121】ステップS E 5では、バッファ(図8(D))の歌詞イベントを、挿入位置PTから挿入行数RG分より後ろのノートプラスデータに割り当てる(図8(F))。以上で、歌詞イベント挿入処理を終了する。

【0122】なお、図8(A)のノートプラスデータに所望の歌詞イベントを挿入して、図8(F)のノートプラスデータを生成する方法は上記以外の方法でもよい。

【0123】図10(A)～(E)は、歌詞イベント削除処理を説明するための図である。図10(A)は、ノートプラスデータの例を示す。操作者は、マウス等を用いて、削除したい歌詞イベントの範囲RGを指定することができます。例えば、範囲RGとして、歌詞イベントL3及びL4を指定することができる。

【0124】まず、図10(B)のクリップボードに、

範囲RG内の歌詞イベントL3及びL4をコピーする。このクリップボードの内容は、この後の処理で使用することはないが、必要に応じて図8(B)のクリップボードとして使用することができる。

【0125】次に、図10(C)のバッファに、範囲RGより後ろの歌詞イベントL5～L8をコピーする。

【0126】次に、図10(D)に示すように、図10(A)の範囲RG以降の歌詞イベントL3～L8を削除する。

【0127】次に、図10(E)に示すように、図10(D)の範囲RG以降のノートイベントN3～N8に、バッファ(図10(C))内の歌詞イベントL5～L8を割り当てる。ただし、歌詞イベントL3及びL4を削除したため、最後のノートイベントN7及びN8には空白行の歌詞イベントが割り当てられる。図10(E)のノートプラスデータでは、歌詞イベントL3及びN4が削除されたため、歌詞イベントL2の次に歌詞イベントL5が続く。

【0128】図11は、上記の歌詞イベント削除処理を実現するためのフローチャートである。

【0129】ステップS F 1では、削除すべき任意行数RGの歌詞イベントを指定し(図10(A))、クリップボード(図10(B))にコピーする。

【0130】ステップS F 2では、削除される行RGより後ろの歌詞イベントをバッファ(図10(C))にコピーする。

【0131】ステップS F 3では、削除される行RG以降の歌詞データを消去する(図10(D))。

【0132】ステップS F 4では、バッファ(図10(C))の歌詞イベントを、削除行RG以降のノートプラスデータに割り当てる(図10(E))。以上で、歌詞イベント削除処理を終了する。

【0133】歌詞イベント削除処理は、演奏データをレンジしてノートイベントの数が少なくなったり、歌詞データの入力ミスにより歌詞イベントの数が多くなった場合等に有効な処理である。

【0134】なお、図10(A)のノートプラスデータから所望の歌詞イベントを削除して図10(E)のノートプラスデータを生成する方法は上記以外の方法でもよい。

【0135】図12(A)～(D)は、第1の歌詞自動割り当て処理を説明するための図である。

【0136】図12(A)は、ノートプラスデータの例を示す。操作者は、マウス等を用いて、自動割り当てを行いたいノートプラスイベントの範囲RGを指定することができます。例えば、範囲RGとして、第2～6のノートプラスイベントを指定することができる。

【0137】まず、図12(B)のバッファに、範囲RG内の歌詞イベント「いうえおかきく」をコピーする。

【0138】次に、図12(C)に示すように、図12

(A) の範囲 RG 内の歌詞イベントを削除する。

【0139】次に、図 12 (D) に示すように、図 12 (D) の範囲 RG 内のノートイベント N2～N6 に、バッファ (図 12 (B)) 内の歌詞イベント「いうえおかきく」を 1 文字ずつ割り当てる。ただし、最後のノートイベント N6 には、残りの文字列「かきく」を割り当てる。図 12 (D) のノートプラスデータでは、範囲 RG 内のノートイベント N2～N5 にそれぞれ 1 文字ずつ割り当たられ、ノートイベント N6 に 3 文字「かきく」が割り当たられる。基本的に、1 音符に 1 文字 (音節) が割り当たられる。

【0140】第 1 の歌詞自動割り当て処理は、範囲 RG の行数内で、すなわち、行数を増減させることなく、各行に自動的に歌詞イベントを割り当てる処理である。

【0141】図 13 は、上記の第 1 の歌詞自動割り当て処理を実現するためのフローチャートである。

【0142】ステップ SG1 では、歌詞を自動割り当てる複数行 RG を選択する (図 12 (A))。

【0143】ステップ SG2 では、選択された複数行 RG に含まれる歌詞イベントをバッファ (図 12 (B)) にコピーする。ただし、歌詞イベントが音素列により構成されているときには、バッファ中の音素列を文字データに変換する。

【0144】ステップ SG3 では、選択された複数行 RG の歌詞イベントを消去する (図 12 (C))。

【0145】ステップ SG4 では、バッファ (図 12 (B)) 内の文字列データを、選択された複数行 RG のノートプラスデータに割り当てる (図 12 (D))。以上で、第 1 の歌詞自動割り当て処理を終了する。

【0146】図 14 (A)、(B) は、空白行自動挿入処理を説明するための図である。図 14 (A) は、ノートプラスデータの例を示す。操作者は、マウス等を用いて、空白行を挿入したい行 RG を選択することができる。例えば、選択行 RG として、第 4 行のノートプラスイベントを指定することができる。

【0147】まず、選択行 RG 内の歌詞イベントを構成する文字数をカウントする。例えば、歌詞イベント「おかげ」の文字数は 4 文字である。

【0148】次に、図 14 (B) に示すように、「当該文字数 - 1」行の空白行を、選択行 RG の位置に挿入する。例えば、 $4 - 1 = 3$  行の空白行を挿入する。選択行 RG 以降のノートイベント N4～N6 には空白行が割り当たられ、ノートイベント N7 に歌詞イベント「おかげ」が割り当たられる。

【0149】なお、選択行 RG の歌詞イベント「おかげ」の前に空白行を挿入する場合に限定されず、歌詞イベント「おかげ」の後ろに空白行を挿入してもよい。

【0150】この後、第 4 行～第 7 行を選択して、図 12 に示した第 1 の歌詞自動割り当て処理を行えば、当該空白行に歌詞イベントを割り当てることができる。具体

的には、「N4, お」、「N5, か」、「N6, き」、「N7, く」のように割り当てることができる。

【0151】図 15 は、上記の空白行自動挿入処理を実現するためのフローチャートである。

【0152】ステップ SH1 では、空白行を自動挿入する行 RG を選択する (図 14 (A))。

【0153】ステップ SH2 では、選択された行 RG の歌詞イベントの文字数を検出する。歌詞イベントが音素列により構成されるときには、音素列を文字に換算してから文字数を検出する。

【0154】ステップ SH3 では、「文字数 - 1」の行数の空白行を選択行 RG の位置に挿入する。この挿入は、上記空白行挿入と同様に、バッファを用いて行うことができる。以上で、空白行自動挿入処理を終了する。

【0155】図 16 (A)～(D) は、第 2 の歌詞自動割り当て処理を説明するための図である。

【0156】図 16 (A) は、ノートプラスデータの例を示す。操作者は、マウス等を用いて、自動割り当てを行いたいノートプラスイベントの範囲 RG を指定することができます。例えば、範囲 RG として、第 2～6 のノートプラスイベントを指定することができます。

【0157】まず、図 16 (B) のバッファに、範囲 RG の歌詞イベント「いうえおかげ」をコピーする。

【0158】次に、図 16 (C) に示すように、バッファ内の文字数 (7 文字) が範囲 RG の行数 (5 行) よりも 2 文字多いので、2 行の空白行を範囲 RG の位置に挿入する。

【0159】次に、図 16 (D) に示すように、選択行 RG の位置からバッファ (図 16 (B)) 内の文字数 (7 文字) までの数のノートイベント N2～N8 に、バッファ内の文字列「いうえおかげ」を 1 文字ずつ割り当てる。ノートイベント N2～N8 には、それぞれ 1 文字の歌詞イベントが割り当たる。

【0160】第 2 の歌詞自動割り当て処理は、選択された行 RG に含まれる全ての文字が、必ず 1 ノートイベントに付き 1 文字割り当たられるようになる処理である。

【0161】図 17 は、上記の第 2 の歌詞自動割り当て処理を実現するためのフローチャートである。

【0162】ステップ ST1 では、歌詞を自動割り当てる複数行 RG のノートプラスイベントを選択する (図 16 (A))。

【0163】ステップ SI2 では、選択された複数行 RG の歌詞イベントをバッファ (図 16 (B)) にコピーする。ただし、歌詞イベントが音素列により構成されているときには、バッファ中の音素列を文字データに変換する。

【0164】ステップ SI3 では、バッファ内の文字数を検出する。ステップ SI4 では、選択された行数 RG と検出された文字数を比較する。

【0165】ステップ SI5 では、比較した行数が等し

いか、多いか、少ないかをチェックする。行数が多いときはステップS16へ進み、余分な行数を削除し、ステップS18へ進む。行数が少ないときにはステップS17へ進み、足りない行数を挿入し(図16(C))、ステップS18へ進む。行数が等しいときには、行数を変更せずに、ステップS18へ進む。

【0166】ステップS18では、バッファ内の文字列データを、選択行RGの先頭行から、選択行RGの行数だけ、又は削除後の行数だけ、又は挿入後の行数だけ、ノートプラスイベントに割り当てる(図16(D))。以上で、第2の歌詞自動割り当て処理を終了する。

【0167】図18(A)～(D)は、複数行併合処理を説明するための図である。図18(A)は、ノートプラスデータの例を示す。ノートプラスデータは、ノートプラスイベント「N1, あ」、「N2, い」、「N3, う」、「N4, え」、「N5, お」を有する。

【0168】操作者は、マウス等を用いて、併合するノートプラスイベントの範囲RGを指定することができる。例えば、範囲RGとして、第2及び3のノートプラスイベントを指定することができる。

【0169】まず、図18(B)のバッファに、範囲RG内の歌詞イベント「いう」をコピーする。

【0170】次に、図18(C)に示すように、範囲RG内の先頭行の歌詞イベント「い」だけを残し、残りの行の歌詞イベント「う」を削除する。範囲RGより後ろの歌詞イベント「え、お」は、前の行に詰められる。ノートプラスデータは、ノートプラスイベント「N1, あ」、「N2, い」、「N3, え」、「N4, お」、「N5, 」を有する。

【0171】次に、図18(D)に示すように、範囲RG内の先頭行のノートイベントN2に、バッファ内の文字列「いう」を割り当てる。第2のノートイベントN2には、範囲RG内の歌詞イベント「いう」が併合されて割り当たられる。

【0172】上記の第1又は第2の歌詞自動割り当て処理では、1つのノートイベントに1文字の歌詞イベントが割り当たが、その後にこの複数行併合処理を利用するにより、1つのノートイベントに2文字以上の歌詞イベントを割り当てるよう修正することができる。

【0173】図19は、上記の複数行併合処理を実現するためのフローチャートである。ステップSJ1では、歌詞イベントを併合する複数行RGを選択する(図18(A))。

【0174】ステップSJ2では、選択された複数行RG内の歌詞イベントをバッファ(図18(B))にコピーする。ただし、歌詞イベントが音素列により構成されているときには、バッファ中の音素列を文字データに変換する。

【0175】ステップSJ3では、選択された複数行RGのうち、1行を残して残りの行の歌詞イベントを削除

する(図18(C))。削除された後の歌詞イベントは、前の行に詰められる。

【0176】ステップSJ4では、バッファ内の文字列データを、範囲RG内の残った行のノートプラスデータ内のノートイベントに割り当てる(図18(D))。以上で、複数行併合処理を終了する。

【0177】図20(A)、(B)は、歌詞イベント分割処理を説明するための図である。図20(A)は、ノートプラスデータの例を示す。ノートプラスデータは、ノートプラスイベント「N1, あ」、「N2, いう」、「N3, え」を有する。

【0178】操作者は、マウス等を用いて、歌詞イベントの分割を行いたい行RGを選択し、その行内で分割を行いたい文字位置にカーソルCSを設定する。例えば、範囲RGとして、第2のノートプラスイベント「N2, いう」を選択し、カーソルCSを「い」と「う」の間に設定する。その後、操作者がリターンキー(実行キー)を操作することにより、以下の分割処理が行われる。

【0179】図20(B)に示すように、文字「い」と「う」とが分割される。ノートプラスデータは、ノートプラスイベント「N1, あ」、「N2, い」、「N3, うえ」を有する。カーソルCSの後ろの文字「う」は、選択行RGの後ろの行の歌詞イベント「え」と併合される。

【0180】空白行の歌詞イベントを多数挿入しておき(図6)、先頭から順番に歌詞を割り当てる際に便利である。

【0181】図21は、上記の歌詞イベント分割処理を実現するためのフローチャートである。

【0182】ステップSK1では、分割する歌詞イベントを含む行RGを選択する(図20(A))。

【0183】ステップSK2では、マウス操作等により分割位置にカーソルCSを設定する(図20(A))。

【0184】ステップSK3では、リターンキーの操作に応じて、分割位置以降の歌詞イベントをその次の行のノートプラスデータ内の歌詞イベントに追加する(図20(B))。以上で、歌詞イベント分割処理を終了する。

【0185】図22(A)～(C)は、仮名変換処理を説明するための図である。図22(A)は、ノートプラスデータの例を示す。ノートプラスデータは、ノートプラスイベント「N1, あか」、「N2, い」、「N3, タ」、「N4, 日」を有する。

【0186】操作者は、マウスを用いて、仮名変換を行いたい歌詞イベント「タ」の位置にマウスポインタ71を移動し、マウスの右ボタンをクリックする。

【0187】図22(B)に示すように、歌詞イベント「タ」の読み仮名候補がメニュー72に表示される。メニュー72には、例えば「ゆう」、「せき」が表示される。操作者は、適切な読み仮名である文字列「ゆう」の

位置にマウスポインタ71を移動させ、マウスの左ボタンをクリックする。

【0188】図22(C)に示すように、ノートイベントN3に割り当てられる歌詞イベントは漢字「夕」から仮名「ゆう」に変換される。漢字の他、英数字や記号等も仮名に変換することができる。漢字や英数字等を仮名に変換すれば、その後に仮名から音素列に変換することができる(図5)、音素列の発音が可能になる。

【0189】図23は、上記の仮名変換処理を実現するためのフローチャートである。ステップSL1では、仮名変換する文字の行をマウスポインタ71で指定する(図22(A))。

【0190】ステップSL2では、マウスの右ボタンがクリックされると、選択文字に対応する読み仮名をデータベースから検索し、読み仮名候補として表示する(図22(B))。データベースは、漢字等から仮名への変換するための辞書であり、外部記憶装置13又はRAM3等に記憶される。

【0191】ステップSL3では、読み仮名候補のいずれかをマウスポインタ71で指定する(図22(B))。

【0192】ステップSL4では、マウスの左ボタンがクリックされると、指定された読み仮名候補を指定行の文字列として、ノートプラスデータ内のノートイベントに割り当てる(図22(C))。以上で、仮名変換処理を終了する。

【0193】以上のようにして作成されたノートプラスイベント(図5の音素列変換処理によって文字列を音素列に変換したもの)を、外部音源装置21へと供給することによって、外部音源装置21内の音声合成回路32において、歌唱音声信号を生成させることができる。なお、本実施の形態において説明した音声データ処理装置(パーソナルコンピュータ1)においては、ノートプラスデータを、そのままの形式で再生処理可能な、いわば、専用の音声データ処理装置である。この音声データ処理装置は、ノートイベントと歌詞イベント(音素列)を適切な順序で送信することができる。

【0194】ここで、ノートイベントと歌詞イベントを送信する際の適切な順序について説明する。図24(A)、(B)は、ノートイベントと歌詞イベントの送信順序を説明するための図である。ノートプラスイベントは、例えばノートイベントN1と歌詞イベントL1の組である(図6(A))。ノートイベントN1と歌詞イベントL1は理論上は同じタイミングで送信すべきであるが、MIDI通信はシリアル通信であるため、いずれを先に外部音源装置21(図1)へ送信すべきかが問題になる。ノートイベントN1はノートオンイベントに変換されて送信される。歌詞イベントは音素列に変換(図5)されて送信される。

【0195】図24(A)は、音素列、ノートオンの順

序で送信するタイミングを示す。まず、音節「あ」に対応する音素列「a j」を送信する。その後にノートオンを送信すると、そのタイミングで「あ」の発音が開始する。次に、音節「い」に対応する音素列「i j」を送信する。その後にノートオンを送信すると、そのタイミングで「あ」から「い」に発音が切り換わる。次に、音節「う」に対応する音素列「u j」を送信する。その後にノートオンを送信すると、そのタイミングで「い」から「う」に発音が切り換わる。

【0196】図24(A)では、ノートオン(ノートイベント)と音素列(歌詞イベント)との対応がとれており、適正な歌を歌わせることができる。

【0197】図24(B)は、ノートオン、音素列の順序で送信するタイミングを示す。まず、ノートオンを送信する。この時点では、音素列が設定されていないので、例えば「あ」等の初期設定されている音素列が発音を開始する。その後、音節「あ」に対応する音素列「a j」を送信する。この時点では、発音に変化はなく初期設定されている「あ」の発音が持続される。次に、ノートオンを送信すると、そのタイミングで初期設定された音素列「あ」の発音から、送信された音素列「あ」の発音に切り換わる。その後に、音節「い」に対応する音素列「i j」を送信するが、発音に影響はない。次に、ノートオンを送信すると、そのタイミングで「あ」から「い」に発音が切り換わる。その後に、音節「う」に対応する音素列「u j」を送信するが、発音に影響はない。

【0198】図24(B)では、ノートオン(ノートイベント)と音素列(歌詞イベント)との対応がとれていず、1つのノート分だけ遅れて歌詞が発音されてしまう。これでは、適正な歌を歌わせることができない。

【0199】外部音源装置21には、図24(A)に示したように、まず音素列を送信し、その後、対応するノートオンを送信する必要がある。上述したように、この実施の形態における音声データ処理装置は、ノートプラスデータを正しく扱うことができる装置であり、つ1のノートプラスイベントについて、先に音素列を送信した後、ノートオンを送信するようになっている。ところが、本音声データ処理装置以外のシーケンサ等においては、ノートプラスイベントを正しく扱うことができない。このため、本音声データ処理装置は、ノートプラスデータを、より汎用的なスタンダードMIDIファイルに変換する機能を備えている。このとき、単純にスタンダードMIDIファイルに変換するだけでは、後述するような不都合が生じてしまう可能性があるため、上述したような送信順序を考慮しての変換をするようしている。以下、ノートイベントと歌詞イベント(音素列)の送信順序を考慮して、ノートプラスデータを、スタンダードMIDIファイルに変換する方法を説明する。

【0200】図25は、ノートプラスデータ75をスタ

ンダードMIDIファイル76に変換する方法を説明するための図である。

【0201】ノートプラスデータ75は、順にノートイベントN1、歌詞イベントL1、デュレーションTT2、ノートイベントN2、歌詞イベントL2、デュレーションTT3を有する。

【0202】ノートイベントN1は、ノートオンイベントNON1とゲートタイムGT1を有する。ノートオンイベントNON1は、例えばノートナンバ(音高)、ベロシティ(音量)を含む。ゲートタイムGT1は、ノートオンからノートオフまでの時間であり、例えば450である。歌詞イベントL1は、音素列により構成される。ただし、呼気記号や漢字等は文字列として歌詞イベントL1内に含まれる。

【0203】ノートイベントN2は、ノートイベントN1と同様に、ノートオンイベントNON2とゲートタイムGT2を有する。ゲートタイムGT2は、例えば220である。歌詞イベントL2は、歌詞イベントL1と同様に、音素列により構成される。

【0204】デュレーションTT2は、ノートオンイベントNON1の発音開始からノートオンイベントNON2の発音開始までの時間であり、例えば480である。デュレーションTT3は、ノートオンイベントNON2の発音開始から次のノートオンイベントの発音開始までの時間であり、例えば240である。ゲートタイムやデュレーションの値は、クロック数で表される。1クロックは、例えば4分音符長の1/480の時間である。

【0205】スタンダードMIDIファイル76は、MIDI規格に適した汎用フォーマットのファイルである。スタンダードMIDIファイル76は、デュレーションとイベントの組で構成される。

【0206】スタンダードMIDIファイル76は、順に、歌詞イベントL1、デュレーションTT1、ノートオンイベントNON1、デュレーションTT2、ノートオフイベントNOFF1、デュレーションTT3、歌詞イベントL2、デュレーションTT4、ノートオンイベントNON2、デュレーションTT5、ノートオフイベントNOFF2、デュレーションTT6を有する。

【0207】歌詞イベントL1及びノートオンイベントNON1は、ノートプラスデータ75中の歌詞イベントL1及びノートオンイベントNON1と同じであるが、並び順が異なる。歌詞イベントL1をノートオンイベントNON1よりも先に送信することにより、図24(A)のように正常な歌を歌わせることができる。

【0208】デュレーションTT1は、歌詞イベントL1の送信からノートオンイベントNON1の送信までの時間であり、例えば5である。歌詞イベントL1をノートオンイベントNON1よりも前のアドレスに位置するのであれば、理論的にはデュレーションTT1は0でもよいと考えることができる。

【0209】しかし、デュレーションTT1を0にすると、スタンダードMIDIファイルを送信するパソコン又はシーケンサによって送信順序が異なることがある。すなわち、デュレーションTT1を0にすると、歌詞イベントL1、ノートオンイベントNON1の順で送信されるとは限らず、ノートオンイベントNON1、歌詞イベントL1の順で送信されることがある。さらに、デュレーションTT1が0でなく、1又は2であるときにも同様な現象が生じる。

【0210】これは、シーケンサ等が歌詞イベントL1とノートオンイベントNON1との重要度を比較し、ノートオンイベントNON1の重要度が高いと判断し、ノートオンイベントNON1を歌詞イベントL1よりも先に送信しているものと考えられる。なお、歌詞イベントは、MIDI規格で決められたシステムエクスクルーシブメッセージにより送信することができる。

【0211】上記の弊害を防止するため、デュレーションTT1を5にする。デュレーションTT1は3以上が好ましいが、値が大きすぎると前のイベントに悪影響を与えることも考えられる。デュレーションTT1は、3~10が好ましい。ただし、これはクロックの音符分解能が4分音符長の1/480の場合であって、音符分解能が異なる場合は上記好ましい数値は他の値をとる。例えば、クロックの音符分解能が4分音符の1/96であれば、1~3程度が好ましい。

【0212】デュレーションTT2は、ノートオンNON1からノートオフNOFF1までの時間であり、ノートプラスデータ75のゲートタイムGT1に相当し、例えば450である。

【0213】ノートプラスデータ75中のデュレーションTT2は、ノートオンNON1から次のノートオンNON2までの時間であり、スタンダードMIDIファイル76中のデュレーションTT2とTT3とTT4とに分解される。

【0214】デュレーションTT2は、上記のように、ノートプラスデータ75のゲートタイムGT1と同じく450である。デュレーションTT4は、歌詞イベントL2の送信からノートオンイベントNON2の送信までの時間であり、デュレーションTT1と同様に5である。デュレーションTT3は、ノートオフイベントNOFF1の送信から歌詞イベントL2の送信までの時間であり、TT3=TT2-TT2-TT4で表される。すなわち、TT3=480-450-5=25である。

【0215】上記のように、歌詞イベントL1をノートオンイベントNON1よりも前のアドレスに位置させ、かつデュレーションTT1を5にすることにより、確実に、歌詞イベントL1をノートオンイベントNON1よりも先に送信することができる。その際、ノートオンイベントNON1のタイミングは変えずに、歌詞イベントL1のタイミングを5だけ早めるので、ノートオンNO

N1による発音タイミングはずれることがない。

【0216】また、ノートプラスデータ75をスタンダードMIDIファイル76に変換することにより汎用性が増し、他のシーケンサ等でも処理することが可能になると共に、スタンダードMIDIファイル76をフロッピディスク等でユーザに供給することが可能になる。

【0217】図26(A)に示すように、スタンダードMIDIファイル(SMF)変換手段81は、ノートプラスデータ75をSMFデータ76に変換することができる。パーソナルコンピュータ1は、SMF変換手段81において、ノートプラスデータ75をスタンダードMIDIファイル76に変換し、外部記憶装置13にセーブすることができる。セーブされたスタンダードMIDIファイル76は、他のパーソナルコンピュータやシーケンサ等で使用することができる。また、パーソナルコンピュータ1がスタンダードMIDIファイル76を外部音源装置21に送信すれば、外部音源装置21はスタンダードMIDIファイル76に応じて発音処理を行う。

【0218】また、図26(B)に示すように、ノートプラス変換手段82は、SMFデータ76をノートプラスデータ75に変換することができる。パーソナルコンピュータ1は、外部記憶装置13に記憶されているスタンダードMIDIファイル76をロードし、ノートプラス変換手段82において、ノートプラスデータ75に変換してRAM3に記憶することができる。この際、上記ノートプラスデータ75をスタンダードMIDIファイル76に変換したのとは逆に、ノートイベントとは独立して記憶されている歌詞イベントを、1つのノートプラスイベントにまとめる処理を行う。まず、スタンダードMIDIファイル中のデータを先頭から順次サーチし、ノートオンイベントが見つかったら対応するノートオフイベントを探してゲートタイムを含む1つのノートオンイベントを生成する。そして、該ノートオンイベントよりも5クロック前までに存在する歌詞イベントをサーチして、見つかった場合は、該歌詞イベントをノートオンイベントに付加してノートオンイベントと歌詞イベントを組にしたノートプラスイベントを生成する。操作者は、上記に示したように、ノートプラスデータ75について、種々の編集を行うことができる。なお、ノートプラスデータ75をスタンダードMIDIファイルに変換する際の処理と同様の処理を、スタンダードMIDIファイル以外の比較的汎用性の高いデータ形式に変換する場合に適用してもよい。

【0219】なお、外部音源装置21内の音声合成回路に音素列を発音させる場合に限定されない。音声合成回路を含む音源ボードをパーソナルコンピュータ1に差し込み、当該音源ボードに音素列を発音させてもよい。その場合は、パーソナルコンピュータ1に外部音源装置21を接続する必要はない。

【0220】本実施例による音声データ処理装置は、パソコンとアプリケーションソフトウェアとによる形態に限らず、電子楽器やシーケンサの形態でもよい。アプリケーションソフトウェアは、磁気ディスク、光ディスク、半導体メモリ等の記憶媒体に記憶させ、パソコンに供給するようにしてもよいし、ネットワークを介して供給するようにしてもよい。

【0221】音声データのフォーマットはスタンダードMIDIファイルのような演奏イベントの発生時刻を1つ前のイベントからの時間で表した「イベント+相対時間」に限定されず、演奏イベントの発生時刻を曲や小節内における絶対時間で表した「イベント+絶対時間」、音符の音高と符長あるいは休符と休符長で演奏データを表した「音高(休符)+符長」、演奏の最小分解能毎にメモリの領域を確保し、演奏イベントの発生する時刻に対応するメモリ領域に演奏イベントを記憶した「ベタ方式」等の形式でもよい。

【0222】音声データは、複数のチャンネルのデータが混在した形式であってもよいし、各チャンネルのデータがトラック毎に別れているような形式であってもよい。

【0223】以上実施例に沿って本発明を説明したが、本発明はこれらに制限されるものではない。例えば、種々の変更、改良、組み合わせ等が可能なことは当業者に自明であろう。

#### 【0224】

【発明の効果】以上説明したように、本発明によれば、仮名又は漢字等を含むテキストデータを処理することができるので、テキストデータの入力又は編集が簡単である。仮名又は漢字が混在した既存のテキストデータを使用することができる。

【0225】また、ノートデータにテキストデータを割り当てる際、カラオケ用のテキストデータにおいて、改行マークと副改行マークの両方を区切りデータとして用いることができる。

【0226】また、音声データを例えばスタンダードMIDIファイル形式で扱うことが可能になるので、汎用性が増すと共に、MIDI通信を行うことができる。

#### 【図面の簡単な説明】

【図1】 パーソナルコンピュータと外部音源装置とが接続されたシステムのハード構成を示す図である。

【図2】 ノートイベントにテキストイベントを割り当てる処理を示す図である。

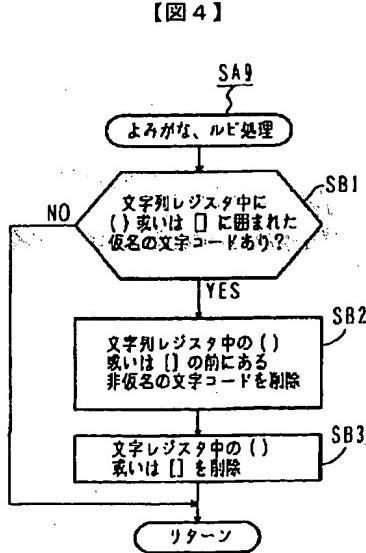
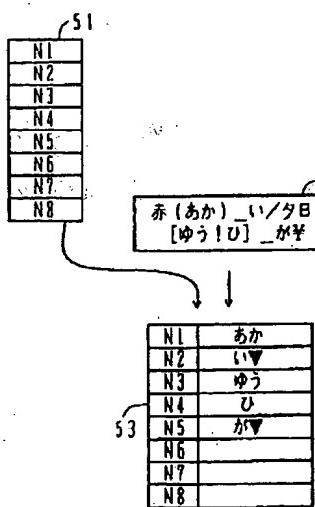
【図3】 テキストインポート処理を示すフローチャートである。

【図4】 図3のステップSA9に示す読み仮名及びルビ処理の詳細を示すフローチャートである。

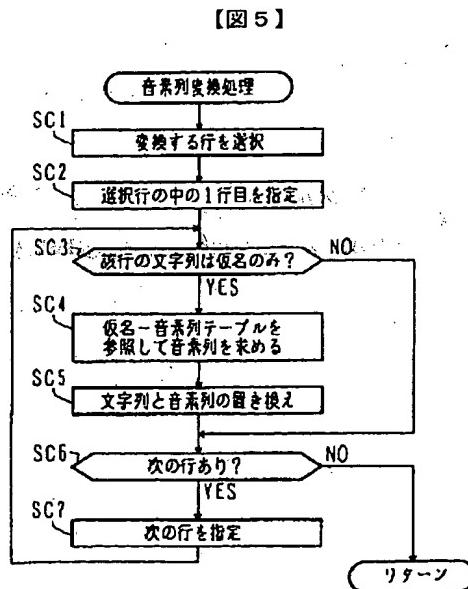
【図5】 音素列変換処理を示すフローチャートである。

【図6】 空白行挿入処理を説明するための図である。

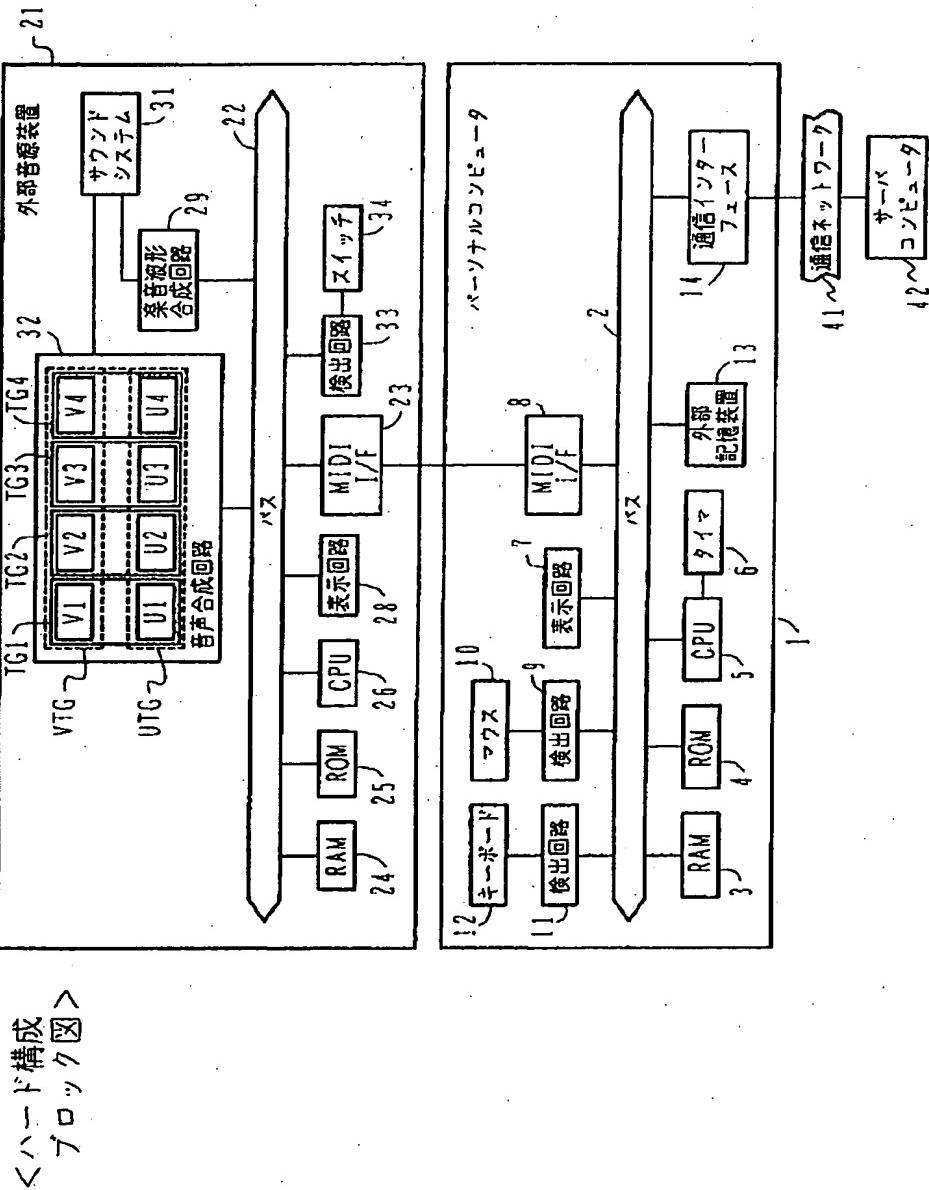
- 【図 7】 空白行挿入処理を示すフローチャートである。
- 【図 8】 歌詞イベント挿入処理を説明するための図である。
- 【図 9】 歌詞イベント挿入処理を示すフローチャートである。
- 【図 10】 歌詞イベント削除処理を説明するための図である。
- 【図 11】 歌詞イベント削除処理を示すフローチャートである。
- 【図 12】 第 1 の歌詞自動割り当て処理を説明するための図である。
- 【図 13】 第 1 の歌詞自動割り当て処理を示すフローチャートである。
- 【図 14】 空白行自動挿入処理を説明するための図である。
- 【図 15】 空白行自動挿入処理を示すフローチャートである。
- 【図 16】 第 2 の歌詞自動割り当て処理を説明するための図である。
- 【図 17】 第 2 の歌詞自動割り当て処理を示すフローチャートである。
- 【図 18】 複数行併合処理を説明するための図である。
- 【図 19】 複数行併合処理を示すフローチャートである。
- 【図 20】 歌詞イベント分割処理を説明するための図である。
- 【図 21】 歌詞イベント分割処理を示すフローチャートである。



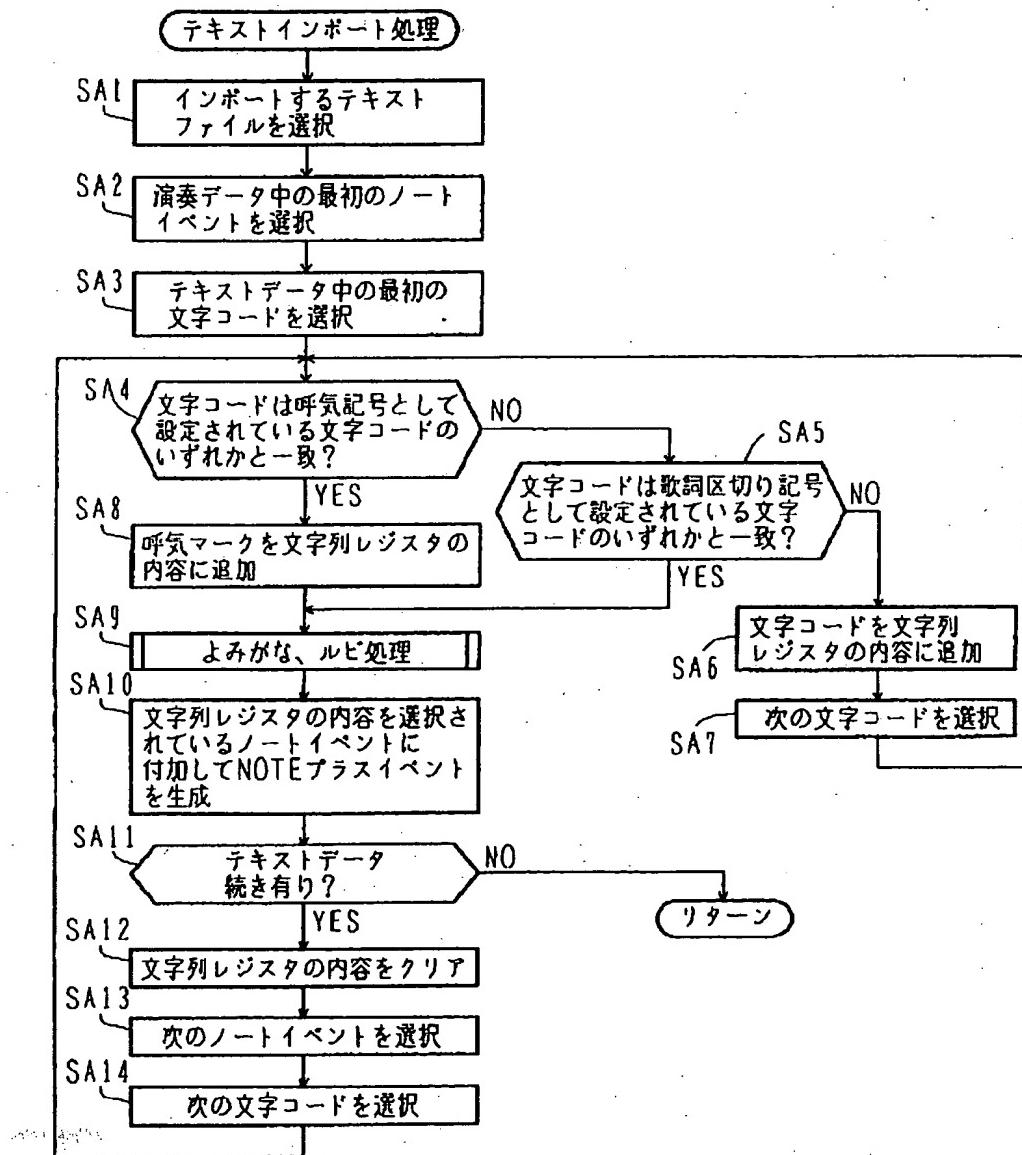
- 【図 22】 仮名変換処理を説明するための図である。
- 【図 23】 仮名変換処理を示すフローチャートである。
- 【図 24】 図 24 (A) は音素列、ノートオンの順序で送信した場合の発音を示し、図 24 (B) はノートオン、音素列の順序で送信した場合の発音を示す図である。
- 【図 25】 ノートプラスデータとスタンダードMIDIファイルとの間の変換を示す図である。
- 【図 26】 図 26 (A) はスタンダードMIDIファイル変換手段を示し、図 26 (B) はノートプラス変換手段を示す図である。
- 【符号の説明】
- 1 パーソナルコンピュータ、 2 バス、 3 RAM、 4 ROM、 5 CPU、 6 タイマ、 7 表示回路、 8 MIDIインターフェース、 9, 11 検出回路、 10 マウス、 12 キーボード、 13 外部記憶装置、 14 通信インターフェース、 21 外部音源装置、 22 バス、 23 MIDIインターフェース、 24 RAM、 25 ROM、 26 CPU、 28 表示回路、 29 楽音波形合成回路、 31 サウンドシステム、 32 音声合成回路、 33 検出回路、 34 スイッチ、 41 通信ネットワーク、 42 サーバコンピュータ、 51 演奏データ、 52 テキストデータ、 53 ノートプラスデータ、 75 ノートプラスデータ、 76 スタンダードMIDIファイル、 81 スタンダードMIDIファイル変換手段、 82 ノートプラス変換手段



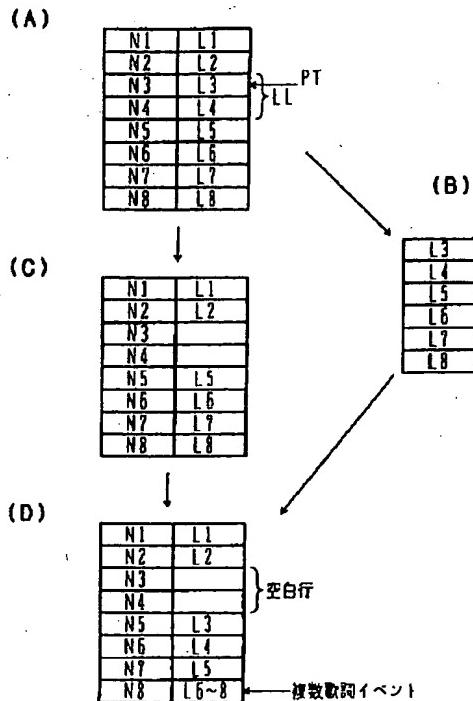
【図1】



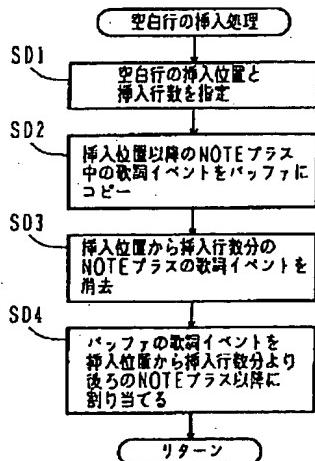
【図3】



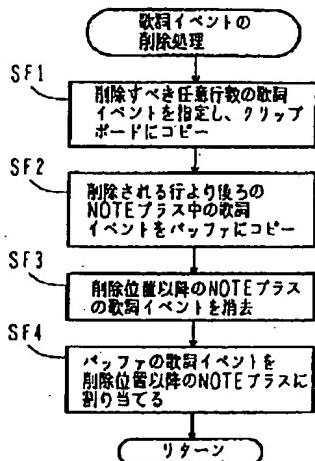
【図6】



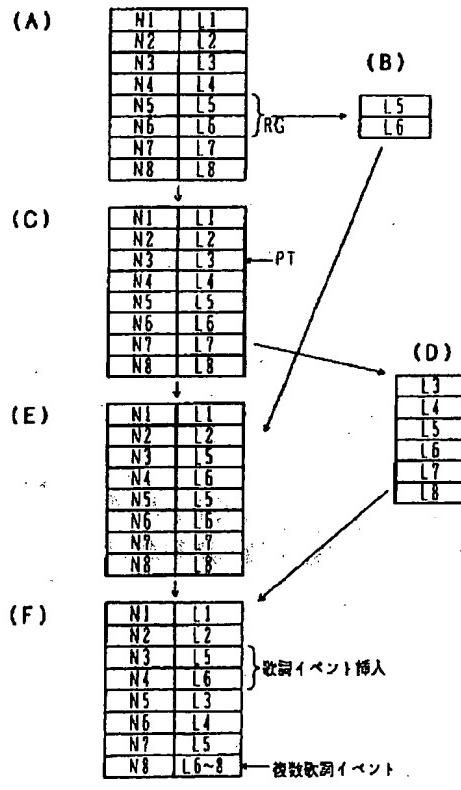
【図7】



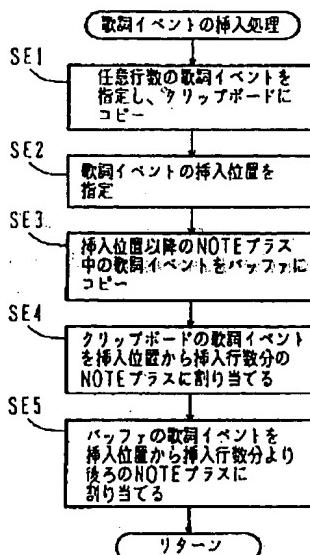
【図11】



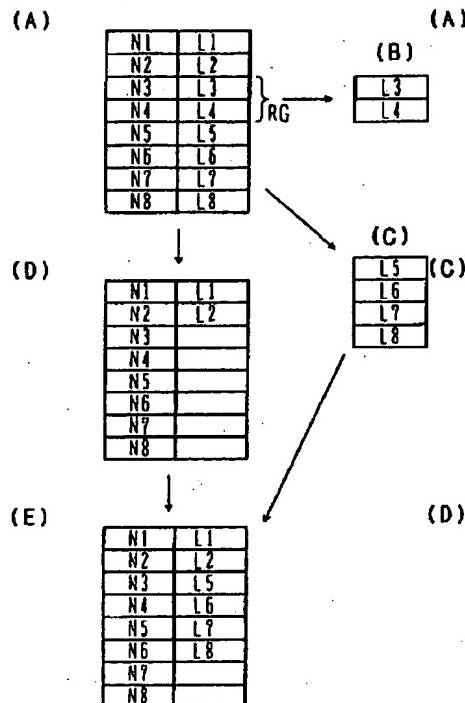
【図8】



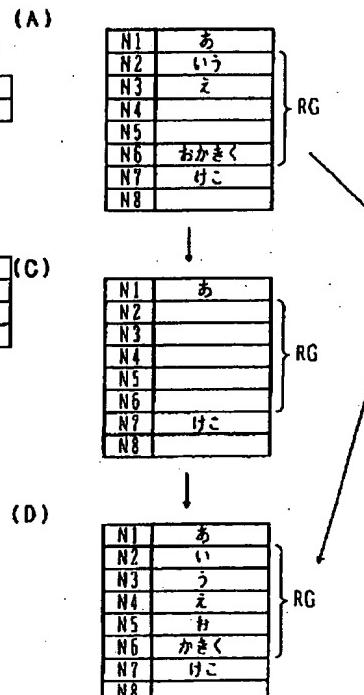
【図9】



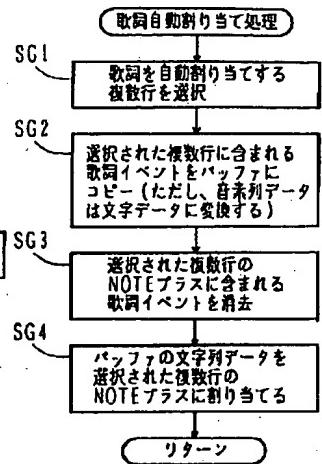
【図 10】



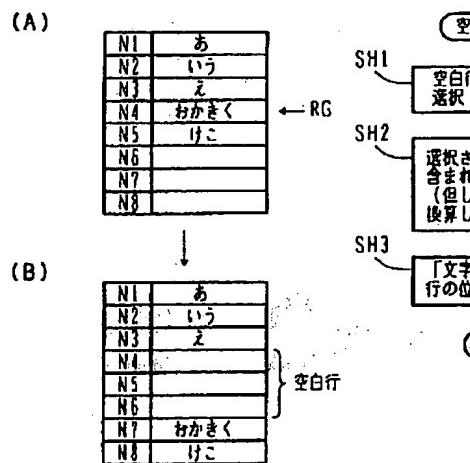
【図 12】



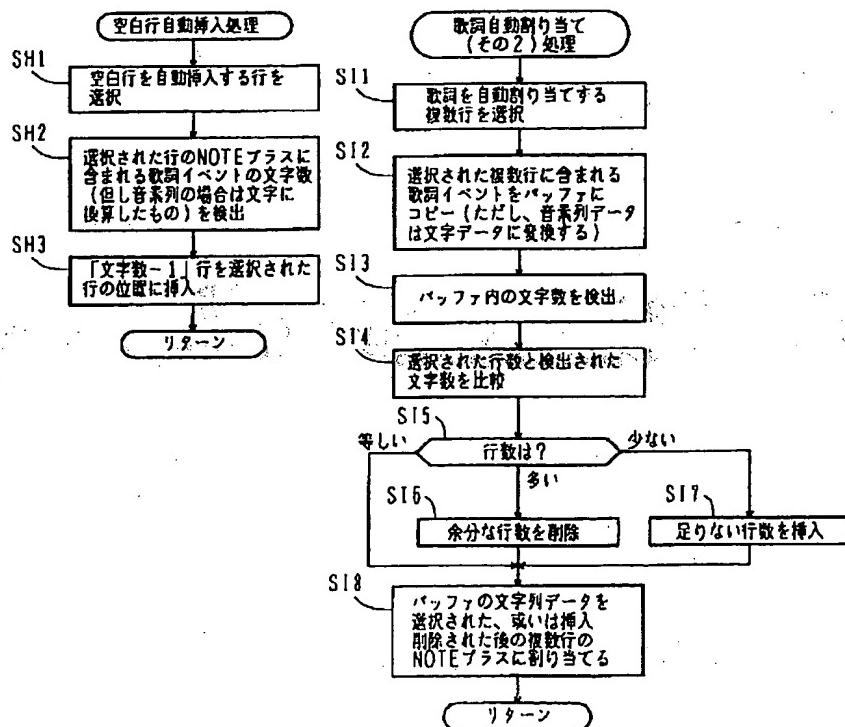
【図 13】



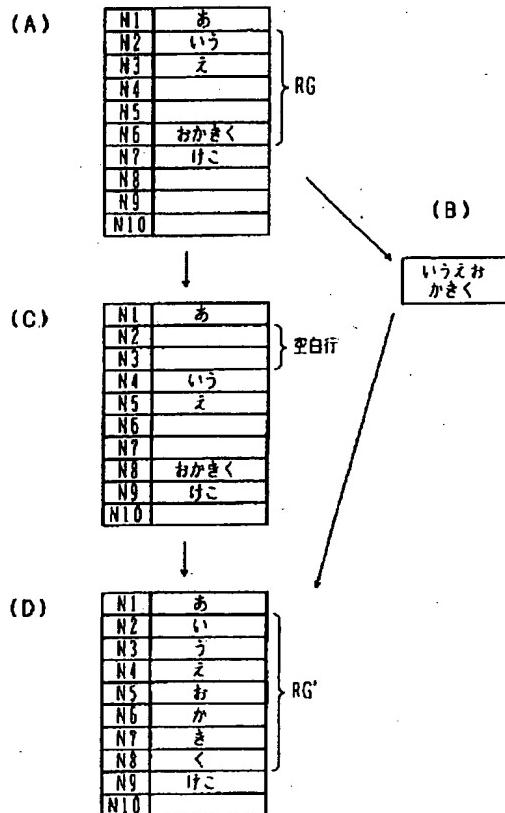
【図 14】



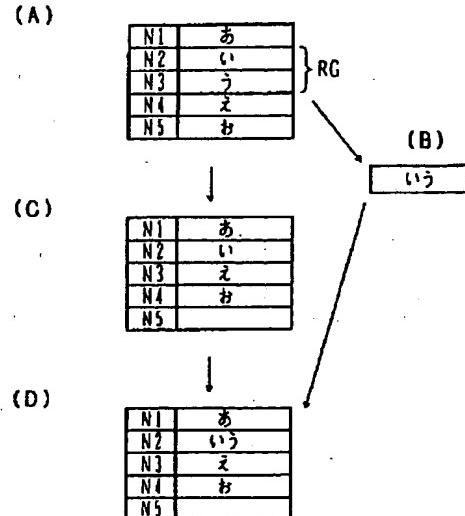
【図 15】



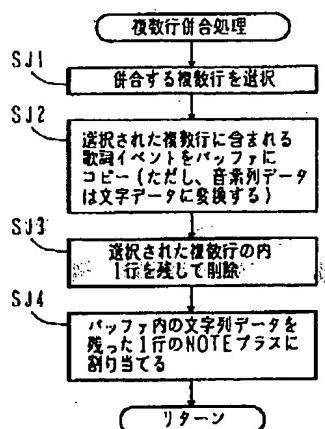
【図 16】



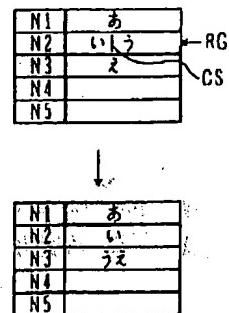
【図 18】



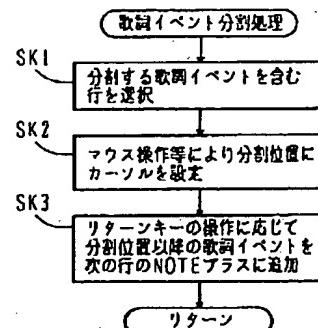
【図 19】



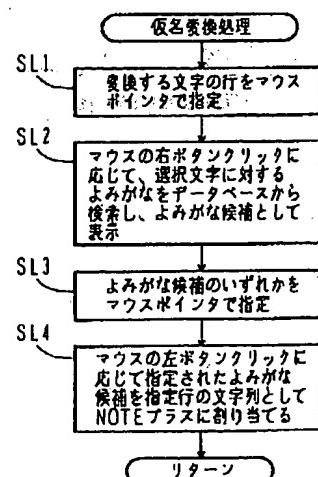
【図 20】



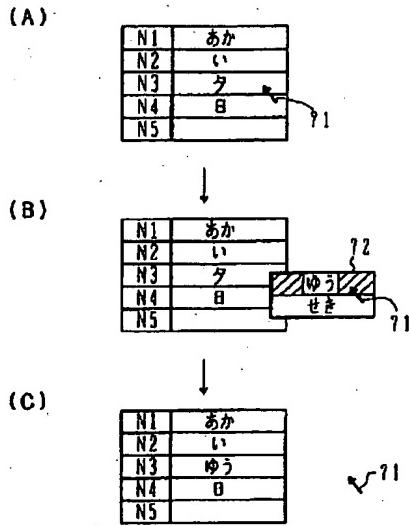
【図 21】



【図 23】

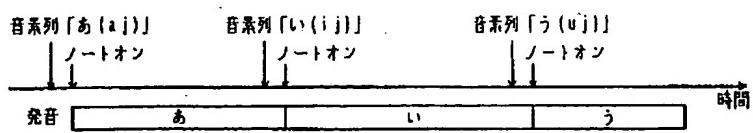


【図 22】

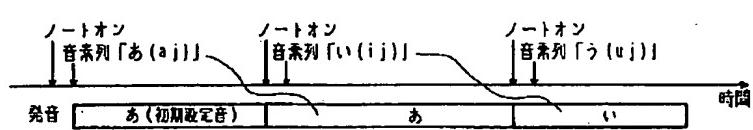


【図 24】

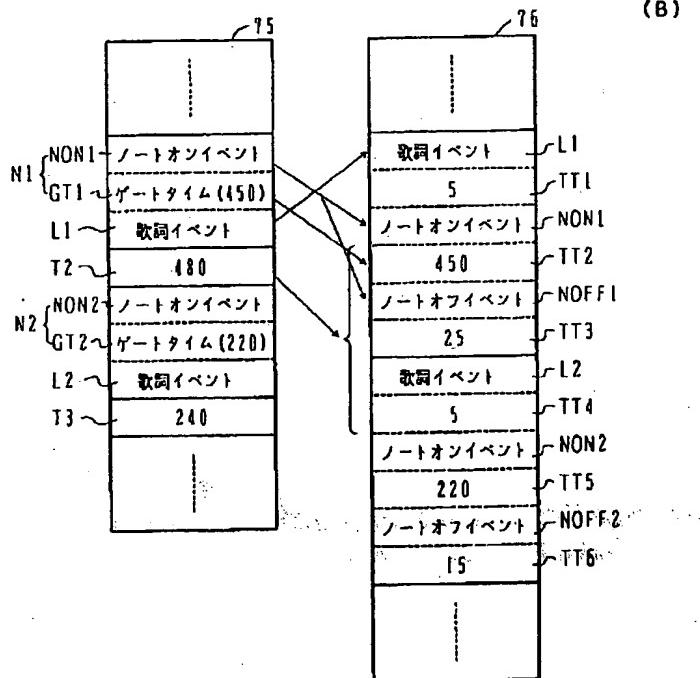
(A) &lt;正規の順序でデータを受信した場合&gt;



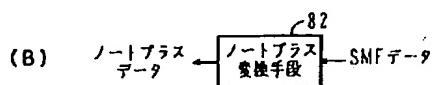
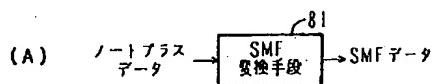
(B) &lt;誤った順序でデータを受信した場合&gt;



【図 25】



【図 26】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**